6-1997

An Ontology Tool for Distributed Information Environments

Kuhanandha Mahalingam

Michael N. Huhns
University of South Carolina - Columbia, huhns@sc.edu

Publication Info

Postprint version. Published in *IEEE Computer*, Volume 30, Issue 6, 1997, pages 80-83.
http://ieeexplore.ieee.org/servlet/opac?punumber=2

An Ontology Tool for Distributed Information Environments[1]

Kuhanandha Mahalingam and Michael N. Huhns
Center for Information Technology
Department of Electrical and Computer Engineering
University of South Carolina
Columbia, SC, U.S.A. 29208
(803) 777-5921 or kuha@sc.edu
(803) 777-8045 FAX

Keywords: Ontology, Java, Healthcare Information Systems, Group Editing, Distributed Database Systems

Abstract

This paper describes how ontologies can be used for query formulation and semantic reconciliation in large distributed information environments. It presents a tool, written in Java, that can be used to create and browse ontologies, and construct ontology-based queries. The tool incorporates several abstraction mechanisms that enable users to manage large ontologies, which are typical of large information environments. The tool is being applied to an information system for healthcare administrators, which spans hospitals, clinics, and governmental health departments.

1. Introduction

The growth and pervasiveness of the Internet and enterprise-wide intranets present unprecedented opportunities for users and application programs to obtain information. Unfortunately, the expanse of such networks presents concomitant problems. The problem of locating and retrieving desired information from the large number of information sources available is exacerbated by both physical and logical heterogeneity:

- The data provided by the sources is no longer just simple text or tuples, but now includes objects and multimedia
- The sources have different policies, procedures, and conventions
- The platforms where the data resides are diverse
- The data have varied and often arcane semantics.

It is further complicated because the networks are growing and evolving as new sources are incorporated or existing sources are revised.

[1] This research was partially done under a cooperative agreement between the National Institute of Standards and Technology Advanced Technology Program (under the HIIT contract, number 70NANB5H1011) and the Healthcare Open Systems and Trials, Inc. consortium.

There are two approaches for coping with these problems. The first, a client-server approach to information management, has produced a plethora of search and query tools that are mostly keyword-based. Keywords are better for text than for the structured data found in most databases, but are completely unsuitable for sources that do not adhere to a uniform semantics, especially the autonomously-maintained databases being deployed widely.

A second approach that achieves interoperation among sources, applications, and users introduces software agents to serve as mediators, translators, and information brokers–this is the essence of a cooperative information systems architecture. The major task for the agents is to reconcile the varied semantics of the mostly autonomous resources.

For either approach, we believe that ontology-based interoperation provides the best solution, especially in environments with heterogeneous semantics. Ontologies can capture both the structure and semantics of information environments. An ontology-based search engine can handle both simple keyword-based queries as well as complex queries on structured data. Reconciliation can be accomplished through the use of a global ontology to which the semantics of the individual resources can be related.

However, the construction of an ontology is itself problematic. We have developed a tool, the Java Ontology Editor (JOE), that not only allows users to build and browse ontologies, but also enables query formulation at several levels of abstraction, including a very abstract level where novice users are more comfortable. This paper describes this tool and its use in a representative application: a healthcare information system [8].

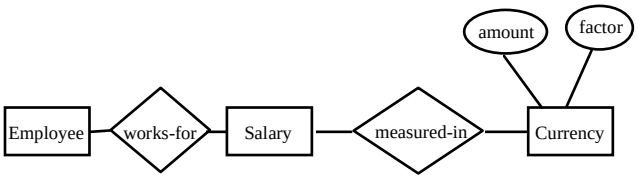
2 Ontologies as a Basis for Interoperation

An ontology is a model of some portion of the world and is described by defining a set of representational terms. Definitions associate the names of entities in a universe of discourse (e.g., classes, relations, functions, or other objects) with formal axioms that constrain the interpretation and well-formed use of these terms [1,2]. Ontologies can organize keywords as well as database concepts by capturing the semantic relationships among the keywords or among the tables and fields in a database. By using these relationships, a network structure can be created providing users with an abstract view of an information space for their domain of interest. Ontologies are well suited for knowledge sharing in a distributed environment, and several ontology-based information systems have been implemented [4-7].

2.1 Advantages for Value Mapping

Ontologies have an advantage over unstructured text-based information spaces for value mapping. Query results do not typically contain information about the units of returned values. For example, when the salaries of employees are requested, the results do not indicate whether the salaries are in dollars or pounds or both. In a heterogeneous information environment, such problems abound.

Figure 1 illustrates an ontology-based solution, where “factor” is used as an attribute of the entity “currency” that is used to measure “salary”. Returned results could contain the appropriate “factor”, which could be used to translate them into the desired currency.



currency name	factor
US $	1.00
DM	0.62
others..	…

Figure 1. Value mapping example using Employee-Salary relationship

2.2 Advantages for Query Formulation

To get desired information in large heterogeneous environments, users must be able to formulate accurate queries efficiently. This is easy if the users are experienced in a database query language, such as SQL, or if a system would allow queries in natural language as, for example, "get me the phone-numbers of all the Johnsons in the Columbia area." We provide a middle ground between natural language and formal query languages: with just a few mouse clicks on the ontology displayed by our tool, the above query would be formulated as

Person
phone-number = <?.request.?>
last-name = 'Johnson'
lives-in
City
name = 'Columbia'

This format is as readable as the English command, and can easily be translated to SQL. Our tool also allows those users who are knowledgeable in SQL to refine these queries if they desire. In addition, users also have the capability to edit and reuse these queries later. This type of query formulation would be difficult without a graphical representation of the ontologies.

2.3 Scalability Advantages

Ontologies can grow and shrink as necessary based on the context where they are being used. In a different context, part of one ontology can be hidden or another made visible, so that a new view of the same information space can be generated to suit a certain audience.

3. Java Ontology Editor (JOE)

JOE is a software tool [3], written in Java, that provides a graphical user interface for creating or editing ontologies, and formulating queries by the point-and-click approach. Queries are formulated on the information space that is displayed by the ontology editor. The use of Java provides advantages in distribution, security, and portability.

3.1 Group Editing

Because Java applets can be downloaded anywhere and anytime, the same ontology can be simultaneously viewed and edited by more than one user. This group editing feature allows people with different expertise to cooperate in creating one global ontology. At the same time, each user can create unique versions for their individual use.

Alternatively, users can merge different individual ontologies to create one large super-ontology.

3.2 Abstraction Mechanisms

Graphical ontology editors typically do not work well with a large number of nodes or links due to the limited viewing area on most computer monitors. In addition, navigating among a large number of nodes and links can be daunting. To overcome this, we added the following abstraction mechanisms to JOE:

- *Selective viewing* - JOE allows the user to view an ontology with complete details or if desired, with only selected types of nodes in the ontology. A user can view only entities, entities and attributes, or entities and relations, greatly reducing the complexity and confusion in a large ontology.

- *Searching* - JOE provides a window, as shown on the left side of Figure 4, with an alphabetic listing of all available nodes in the ontology. The user can locate a node by just double clicking on its name in the list, and JOE will scroll the viewing window to that node. This option minimizes the tedium in searching through a large graph.

- *Zooming* - JOE can display an entire ontology inside the current window by zooming out appropriately. This feature is shown in Figure 2 for a large healthcare ontology.

- *Magnification* - When the entire ontology is displayed, JOE also provides a "magnifying glass" that will magnify a small portion of the ontology under the mouse. This is necessary if the ontology is quite large and detailed information cannot be displayed in the zoomed-out image of the ontology. This magnified portion is displayed in a separate window, as shown in the right side of Figure 2. Clicking anywhere inside the window sets the viewing mode to normal and centers the window at that point in the ontology.

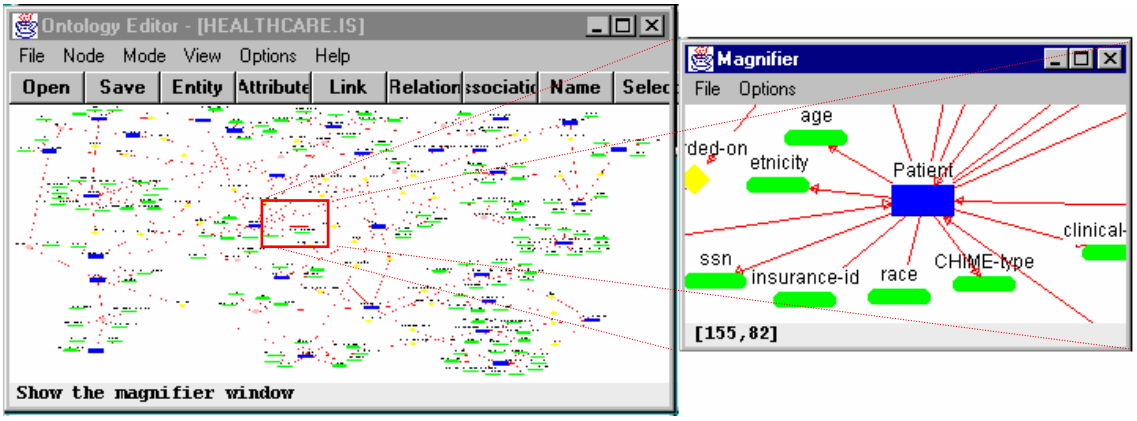


Figure 2. JOE displaying entire ontology within the current window and a magnified view of the selected area on the right

- *Hierarchical information* - JOE can display the hierarchical information of a given ontology in a tree-like structure (similar to the MS Windows file manager format). As shown in Figure 3, JOE will display only the expanded nodes and not the others. With this feature, users can selectively view or work at the level of abstraction they feel comfortable.

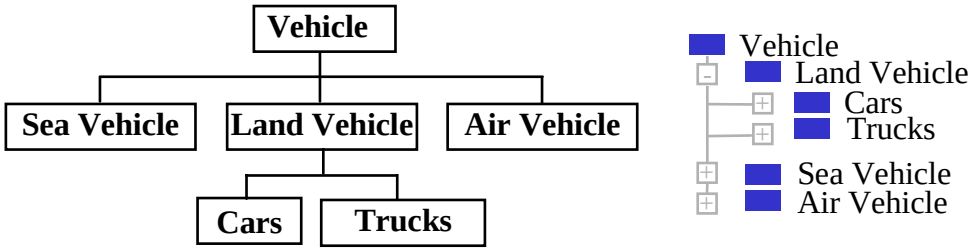


Figure 3. A vehicle ontology and its corresponding hierarchy tree

In addition to above mentioned abstraction mechanisms, JOE also supports most basic editing functionality, such as selecting, cutting, and moving.

4. A Test Application

Currently, JOE is being integrated with InfoSleuth [6]—an agent-based information technology architecture—and applied to a healthcare information system. The healthcare industry provides many opportunities in which such tools as JOE can be used. Although there is a need for healthcare institutions to share information, they cannot request information directly from each other, because there are no globally accepted semantics.

The idea behind our application is to represent, simply, the abstract view of the information fundamental to all healthcare industries by a global ontology, so that queries can be made on this ontology in a standard manner. These queries would be further refined through intermediate "translating-agents" within InfoSleuth, before being processed by individual healthcare providers. All healthcare providers will be able to communicate freely with each other, while continuing to maintain their individual information source architectures. This is not only a feasible solution but also an economical one, since the cost involved in reengineering each facility to adhere to a new standard would be very expensive.

Figure 4 shows JOE executing in its editor mode. A part of a healthcare facility information space is displayed as a graph showing the Patient table, all of its columns, and a few of its relations. A user can move to any node in the information space by selecting from a list of objects, attributes, and relations, shown in the left window.

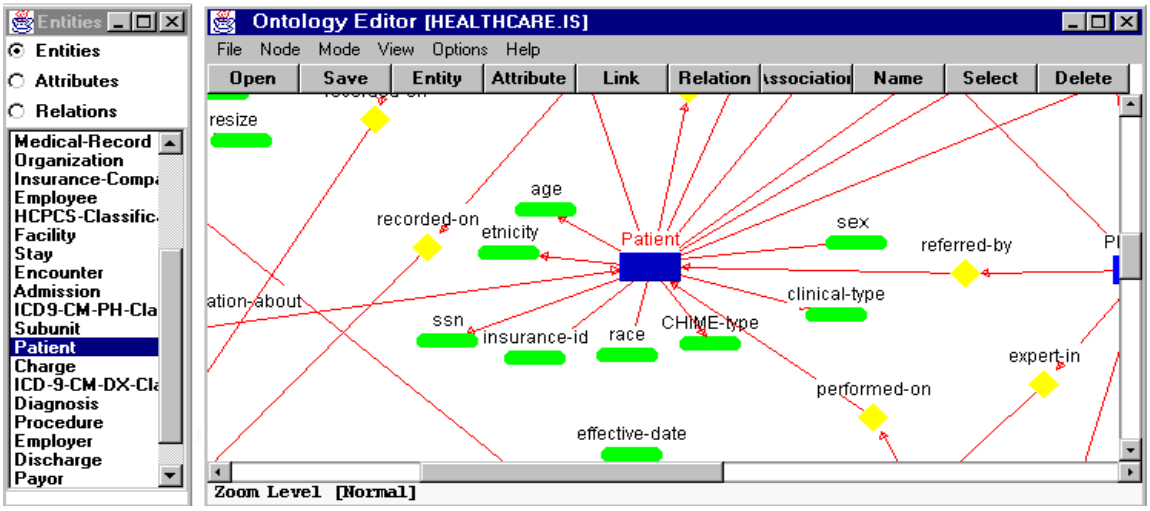


Figure 4. The editor mode of JOE

When executing in the query mode, JOE initially provides an option to select a "root-entity (table)" from all available entities in the currently displayed ontology. Once the root-entity has been selected, the user can build a query by any of the following combinations:

- Clicking on the attributes (shown as ovals)—a dialog box will be displayed where the user can set attribute constraints.
- Clicking on a relation (shown as a diamond) or association (shown as a pentagon)—all the entities related by this relation will be displayed for further expansion.

The current query will be displayed on a separate window to the right of the main window. When complete, the query can be run by choosing the "submit" option in the "Query" menu and the results will be displayed on a separate window. JOE also provides an editor where a user can directly modify or type in a new SQL statement for execution.

5. Conclusions

We believe that an ontology-based representation of information spaces can be effectively used to achieve interoperation among distributed and heterogeneous information sources. Ontologies can capture the structure and semantics of information sources, and are the best tools for semantic reconciliation among software agents. Our tool, JOE, is an attempt to provide an adaptable software agent written in Java to create and edit such ontologies. JOE also provides a unique and easy to use query formulation tool that can function as an independent agent and has the ability to communicate with other agents via Java's RMI protocol.

Through JOE, human and software agents can use or modify the ontologies from anywhere on the Internet. By use of Java's security features, access to such ontologies can be monitored to avoid misuse. Those users who have access to these global ontologies can make queries on them by use of point-and-click approaches. It is our intention that JOE be intuitive enough so that all users, regardless of their expertise in a given area, should be able to get desired information in an efficient manner.

References

[1] T. Gruber. “A translation approach to portable ontologies,” *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[2] P. D. Karp, K. Myers, and T. Gruber, "The generic frame protocol," in *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, pp. 768--774, 1995.

[3] K. Mahalingam, “The Java Ontology Editor (JOE),” Center for Information Technology, Electrical and Computer Engineering Department, University of South Carolina, 1996.

[4] D. Lenat and R. V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1990.

[5] M. Genesereth. “Knowledge Interchange Format,” *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, Cambridge, MA, pages 599-600, Morgan Kaufmann, 1991.

[6] D. Woelk, M. Huhns, and C. Tomlinson. “Uncovering the Next Generation of Active Objects,” *Object Magazine*, vol. 5, no. 4, pp. 32--40, July/August 1995.

[7] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaratnam, Y. Sagiv, Ullman, and J. Widom, “The TSIMMIS Approach to Mediation: Data Models and Languages,” *Proceedings of the NGITS (Next Generation Information Technologies and Systems)*, June 1995.

[8] The Healthcare Information Infrastructure Technology Program *http://host.scra.org/hiit.html*

Kuhanandha Mahalingam is a Ph.D. student in the Electrical and Computer Engineering Department at the University of South Carolina. His research interests include ontology-based distributed information systems and software agent technology. He received his BS and MSEE degrees from North Carolina State University. He is a member of IEEE and NSPE.

Michael Huhns is a professor of electrical and computer engineering and director of the Center for Information Technology at the University of South Carolina. Prior to this he was a senior researcher at the Microelectronics and Computer Technology Corporation (MCC) and an adjunct professor in computer sciences at the University of Texas, Austin. Dr. Huhns received the B.S.E.E. degree in 1969 from the University of Michigan, Ann Arbor, and the M.S. and Ph.D. degrees in electrical engineering in 1971 and 1975, respectively, from the University of Southern California, Los Angeles.

Dr. Huhns is a member of Sigma Xi, Tau Beta Pi, Eta Kappa Nu, ACM, IEEE, and AAAI. He is the author of over 100 technical papers in machine intelligence and an editor of the books *Distributed Artificial Intelligence, Volumes I and II*. His research interests are in the areas of DAI, multiagent systems, and heterogeneous distributed databases. Dr. Huhns is an associate editor for IEEE Internet Computing, IEEE Expert,

and the ACM Transactions on Information Systems. He is on the editorial boards of the Journal of Intelligent Manufacturing and the International Journal on Cooperative Information Systems.